REMARKS

In the patent application, claims 1-36 are pending. In the office action, all pending claims are rejected.

Applicant has amended claims 1 and 11-32.

Claims 1, 11, 20 and 26 have been amended to include the limitation that the client provides information indicative of the one parameter to the server. The support for the amendment can be found on p.12, lines 22-23; p.15, lines 26-27 and p.16, lines 13-14 of the specification.

Claims 1, 21, 26 and 32 have been amended to move part of the preamble to the characteristic part.

Claims 11-20 have been amended to change "multimedia streaming system" to "multimedia streaming network".

Claims 21-25 have been amended to claim a computer readable medium, instead of a software application product.

Claims 26-31 have been amended to claim an apparatus, instead of a terminal in a multimedia streaming network.

No new matter has been introduced.

At section 2, claims 1-36 are rejected under 35 U.S.C. 102(e) as being anticipated by *Bo et al.* (U.S. Patent Application Publication No. 2004/0098748, hereafter referred to as *Bo*).

At section 3, claims 1-36 are rejected under 35 U.S.C. 102(e) as being anticipated by *Bo et al.* (U.S. Patent Application Publication No. 2004/0098748, hereafter referred to as *Bo*). Applicant respectfully disagrees.

<u>A. *BO* FAILS TO ANTICIPATE CLAIMS 1, 11, 21, 26 AND 23</u>

It is respectfully submitted that claims 1, 11, 21, 26 and 32 have the limitations of

defining in the client at least one parameter for determining a rate adaptation operating range so as to carry out rate adaptation between the server and the client;

providing the server information indicative of said at least one parameter;

adapting in the server the data amount to a reception rate based on said at least one parameter;

adjusting in the client packet transfer delay variation based on said adapting.

<u>The Cited *Bo* Reference</u>

Bo discloses an MPEG-4 live uni-cast video streaming system having a server to send data packets to a client using RTP (Real-time Transport Protocol)/UDP (User Data Protocol) protocol in a wireless network. The server receives live video data from a rate-adaptive MPEG-4 Simple Profile encoder. The server has a data transmission module for segmentizing the live video data on the boundary of GOV (group of video object planes) and packetizing each GOV as the payload of the RTP packets. The data transmission pushes the RTP packets to the client through a wireless network according to each GOV data bitrate. The client has a rate-adaptive MPEG-4 Simple Profile decoder to decode the received packets in order to get decoded pictures. The client also has a data link buffer to store the GOV data and to monitor the buffering status. *See* paragraphs [0063] to [0065].

The client has a bitrate adapter module (Figure 2, block 27). The server also has a bitrate adapter module (Figure 2, block 23). The data link buffer in the client collects its buffering status and forwards that information to the client's bitrate adapter module. *See* paragraph [0077]. Based on the information, the client's bitrate adapter module evaluates the bitrate control information and sends it to the server's bitrate adapter module. *See* paragraph [0078]. Based on the received bitrate control information, the server's bitrate adapter module makes a decision on bitrate adjustment and sends a command to the rate-adaptive MPEG-4 Simple Profile encoder so as to allow the encoder to adjust the next GOV's encoding bitrate. *See* paragraph [0079].

Thus, in order for the client to request a change in the encoding bitrate, the client sends **<u>bitrate control information</u>** to the server so as to allow the server to make a decision on bitrate adjustment. *Bo* discloses that the bitrate control information comprises the <u>collected buffer state information</u> ([0065], [0092]). *Bo* does not specifically define what the collected buffer state information is, but it can be construed that the collected buffer state information is the <u>buffering</u>

status of the data link buffer ([0065], [0092]). *Bo* also discloses that the bitrate control information comprises statistical information of the packet loss rate as collected by the data link buffer ([0157], [0158]). In Figure 19, *Bo* discloses sending out bitrate control information at step S147 within the client ([0176]). In Figure 20, *Bo* discloses that the bitrate control information is only sent from the data link buffer in the client to the bitrate adapter in the client so as to allow the change the bitrate.

Bo does not specifically disclose defining in the client at least one parameter for determining a rate adaptation rang and providing informative indicative of that parameter to the server. *Bo* does not disclose or suggest adjusting in the client packet transfer delay variation.

(i) *Bo* fails to disclose defining in the client at least one parameter for determining a rate adaptation operating range and providing information indicative of the parameter

In the office action, the Examiner states that *Bo* discloses defining in the client 12 at least one parameter (network condition) for determining a rate adaptation operating range (paragraphs [0060], [0092]-[0095], [0110].

Applicant respectfully disagrees.

Bo discloses that the bitrate adaptation to the available network bandwidth consists of two aspects:

1. Decrease of the encoding bitrate due to network deterioration or decoder's poor throughput, and
2. Increase of the encoding bitrate due to the health of the network condition ([0069]-[0071]; [0096]-[0098]).

In paragraph [0092], Bo discloses that the bitrate adapter module 27 implements the bitrate adaptation protocol and the network bandwidth polling protocol to feedback bitrate control information to the streaming server 11. As mentioned earlier, *Bo* discloses that the bitrate control information comprises the collected buffer state information ([0065], [0092]). *Bo* does not specifically define what the collected buffer state information is, but it can be construed that the collected buffer state information is the buffering status of the data link buffer ([0065],

[0092]). *Bo* also discloses that the bitrate control information comprises statistical information of the packet loss rate as collected by the data link buffer ([0157], [0158]).

In paragraph [0110], Bo discloses that during the session, both the RTSP server module 21 and the RTSP client module 25 sense the counterpart's statuses by exchanging GET_PARAMETER messages via the wireless network 15. The GET_PARAMETER messages act as keep-alive messages. At the end of the session, the RTSP client module 25 sends the tear-down message to the RTSP server module 21. The RTSP server module 21 terminates the session in accordance with the tear-down message. The keep-alive messages and tear-down messages have nothing to with rate adaption.

Furthermore, *Bo* does not disclose sending a parameter indicative of the network condition to the server.

(ii) *Bo* fails to disclose adjusting in the client packet transfer delay variation

The Examiner also states that *Bo* discloses adjusting in the client packet transfer delay variation (rate adaptation 12 and 27, [0073]-[0075]). Applicant respectfully disagrees.

Bo only discloses having a bitrate adapter module 27 in the client 12. The bitrate adapter module 27 can be used to implement the bitrate adaptation protocol and the network bandwidth polling protocol to feedback bitrate control information to the streaming server 11 ([0092]). The bitrate adapter module 27 can also be used to negotiate with the bitrate adapter module 23 in the client the initial streaming bitrate and how far the current network bandwidth is over the current streaming bitrate ([0107)].

Bo also discloses how the client reconstructs the GOV in case of a packet loss during the transmission of packets as follows:

> [0073] Preferably, if the data transmission module in the client receives the incoming RTP packets (RTP/UDP packets), it starts the reconstruction of each GOV, in other words, each access unit. Then, the recovered GOV is inserted into the data link buffer in the client.
>
> [0074] Preferably, if a packet loss occurs during the transmission of RTP/UDP packets, at least one blank GOV or at least one partially recovered GOV is inserted into the data link buffer in the client.

[0075] Preferably, the data link buffer in the client checks whether it is necessary to retransmit a GOV or a part of a GOV. The retransmission checking is triggered by the insertion of a fully recovered GOV. The data link buffer generates retransmission requests in accordance with the results of the retransmission checking. The retransmission requests are passed from the data link buffer to the data transmission module in the client, and are then transmitted to the streaming server by RTCP/UDP packets propagating through the wireless network. In this case, it is desirable to try the retransmission of a GOV or a part of a GOV only once. Only GOVs that are still in the data link buffer within the streaming server can be retransmitted.

How to deal with a packet loss, however, has **nothing** to do with rate adaptation, nor with adjusting packet transfer delay variation.

For the above reason, *Bo* fails to anticipate claims 1, 11, 21, 26 and 32.

B. *BO* FAILS TO ANTICIPATE CLAIMS 2-10, 12-20, 22-25, 27-31 AND 33-36

(i) *Bo* fails to disclose that the parameter comprises a minimum shift amount indicative of a difference between a sampling time and a transmission time of a packet at the server

In rejecting claim 2, the Examiner states that *Bo* discloses that the parameter comprises a minimum shift amount indicative of a difference between a sampling time and a transmission time of a packet at the server so as to allow the server to carry out said adapting based on the minimum shift amount ([0107], [0110]).

At paragraph [0107], *Bo* discloses:

The bitrate adapter module 27 in the client 12 and the bitrate adapter module 23 in the streaming server 11 negotiate the initial streaming bitrate using the network bandwidth polling protocol by temporarily opening a UDP connection via the wireless network 15. In this case, the network bandwidth polling process can be triggered by a polling timer during the data streaming procedure. The bitrate adapter module 27 in the client 12 and the bitrate adapter module 23 in the

streaming server 11 negotiate how far the current network bandwidth is over the current streaming bitrate by temporarily opening a UDP connection via the wireless network 15.

This paragraph only describes how the client and the server negotiate the initial bitrate and what the maximum bitrate difference between the current streaming bitrate and the network bandwidth is allowed.

At paragraph [0110], *Bo* discloses:

> [0110] The RTSP client module 25 initially sends the setup message to the RTSP server module 21 via the wireless network 15 to ask for a session setup. When the RTSP client module 25 gets an active ACK (acknowledgment), it creates the object of the RTP/RTCP transport engine client 26 and sends the play message to the RTSP server module 21 via the wireless network 15 to ask for start of the streaming. The RTSP server module 21 then controls the RTP/RTCP transport engine server 22 to provide the streaming service. During the session, both the RTSP server module 21 and the RTSP client module 25 sense the counterpart's statuses by exchanging GET_PARAMETER messages via the wireless network 15. The GET_PARAMETER messages act as keep-alive messages. At the end of the session, the RTSP client module 25 sends the tear-down message to the RTSP server module 21. The RTSP server module 21 terminates the session in accordance with the tear-down message.

This paragraph only describes the RTSP session procedure between the server and the client as shown in Figure 3. RTSP (real-time streaming protocol, RFC2326) modules are only used for <u>session</u> control. This has <u>**nothing**</u> to do with sampling time. These two cited paragraphs have <u>**nothing**</u> to do with the difference between the sampling time and the transmission time of the packet.

<u>(ii) *Bo* fails to disclose that the parameter comprises a target shift amount indicative of a shift amount greater than a difference between a sampling time and a transmission time of a packet at the server</u>

In rejecting claim 3, the Examiner states that *Bo* discloses the parameter comprises a target shift amount indicative of a shift amount greater than a difference between a sampling time and a transmission time of a packet at the server so as to allow the server to carry out said adapting based on the target shift amount ([0170], [0110]).

Again, the RTSP session procedure has **nothing** to do with the difference between the sampling time and transmission time of a packet.

(iii) *Bo* fails to disclose that the parameter comprises a number specifying a maximum difference between the number of bytes that has been sent and the number of bytes that have been sampled

In rejecting claim 4, the Examiner states that the parameter comprises a number specifying a maximum difference between the number of bytes that has been sent and the number of bytes that have been sampled so as to allow the server to carry out said adapting based on the number ([0107], [0110]).

The RSTP session procedure has **nothing** to with the number of bytes that has been sent and the number of bytes that have been sampled.

(iv) *Bo* fails to disclose adapting a sampling rate to the transmission rate in the server based on the parameter

In rejecting claim 5, the Examiner states that *Bo* discloses adapting a sampling rate to the transmission rate in the server based on the parameter (network connection; [0092]-[0095]).

It is not clear how network connection has anything to do with adapting the sampling rate to the transmission rate in the server.

Furthermore, in the following paragraphs *Bo* discloses:

[0092] The client 12 receives the MPEG-4 simple profile live video data from the streaming server 11 through the wireless network 15. The client has an RTSP (real-time streaming protocol, RFC2326) client module 25 which performs session control. The RTSP client module 25 in the client 12 and the RTSP server module 21 in the streaming server 11 are counterparts with respect

to each other. The client 12 has a data transmission module 26 constituting an RTP/RTCP (real-time transport protocol/real-time control protocol) transport engine client. The data transmission module 26 receives the RTP (real-time transport protocol, RFC1889) packets from the streaming server 11 through the wireless network 15. The data transmission module 26 depacketizes MPEG-4 data blocks from the payload of the received RTP packets, and desegmentizes the MPEG-4 data blocks back to an original GOV (group of video object planes) referred to as a recovered GOV. The data transmission module 26 reconstructs an MPEG-4 video stream composed of recovered GOVs, whereas RTCP (real-time control protocol, RFC1889) is implemented to send the retransmission request. The client 12 has a bitrate adapter module 27. The bitrate adapter module 27 implements the bitrate adaptation protocol and the network bandwidth polling protocol to feedback bitrate control information to the streaming server 11. The bitrate adapter module 27 in the client 12 and the bitrate adapter module 23 in the streaming server 11 are counterparts with respect to each other. The client 12 has a data link buffer 28 connected among the data transmission module 26, the bitrate adapter module 27, and the MPEG-4 decoder 13. The data link buffer 28 stores the GOV data (the reconstructed MPEG-4 video stream) and monitors the buffering status of itself as well as forwards the collected buffer state information to the bitrate adapter module 27 as the bitrate control information.

[0093] The initial streaming bitrate is decided by two ways (1) and (2) as follows.

[0094] (1) Manually configured at the client 12 by the user through a GUI (graphic user interface); and

[0095] (2) Auto-negotiated by the streaming server 11 and the client 12 with the network bandwidth polling protocol.

In paragraph [0092], *Bo* only describes what the client is adapted to do when it receives video data from the server 11. This paragraph has **nothing** to do with adapting the sampling rate to the transmission rate in the server.

In paragraphs [0093] to [0095], *Bo* only describes ho the initial streaming bitrate is decided. The initial streaming bitrate has **nothing** to do with adapting the sampling rate to the transmission rate in the server.

(v) <u>*Bo* fails to disclose that the parameter comprises a clock shift amount for preventing playout disruption in the client</u>

In rejecting claim 6, the Examiner states that *Bo* discloses that the parameter comprises a clock shift amount for preventing playout disruption in the client ([0114]-[0116]; [0128]-[0135]).

In the following paragraphs, *Bo* discloses:

[0114] FIG. 5 schematically shows the overview of the normal data transmission between the RTP/RTCP transport engine server 22 and the RTP/RTCP transport engine client 26. With reference to FIGS. 2 and 5, at the streaming server 11, each raw GOV fed from the MPEG-4 encoder 10 is inserted into the data link buffer 24 with related information such as a GOV bitrate, a GOV duration, and a GOV size. In the data link buffer 24, a new memory node is allocated to store the newly inserted GOV with its related information. The RTP/RTCP transport engine server 22 takes one GOV node from the data link buffer 24. The RTP/RTCP transport engine server 22 calculates the RTP packet duration according to the GOV bitrate, the GOV size, and the RTP packet size. The RTP/RTCP transport engine server 22 sets the push timer 33 in response to the calculated RTP packet duration. When the push timer 33 triggers, an extended RTP packet (a fragment of GOV) is sent from the RTP/RTCP transport engine server 22 to the client 12. The RTP/RTCP transport engine server 22 dispatches RTP packets at intervals decided by a msec timer according to the GOV bitrate. The msec timer is provided by the push timer 33.

[0115] At the client 12, the RTP/RTCP transport engine client 26 receives RTP packets containing GOV data. The RTP/RTCP transport engine client 26 tries to reconstruct the GOV. The RTP/RTCP transport engine client 26 inserts the successfully recovered GOV into the data link buffer 28. The data link buffer 28 checks for any GOV that needs to be retransmitted wholly or partially, and generates a corresponding request (retransmission request). The data link buffer 28 feeds the retransmission request to the RTP/RTCP transport engine client 26. Retransmission requests are transmitted between the RTP/RTCP transport engine client 26 and the RTP/RTCP transport engine server 22.

[0116] FIG. 6 schematically shows the overview of the data retransmission between the RTP/RTCP transport engine server 22 and the RTP/RTCP transport engine client 26. In general,

there are two types of retransmission, that is, the retransmission of a single RTP packet and the retransmission of a whole GOV. The corresponding requests (retransmission requests) are handled by the CRTCP sections 31 and 35, which realize the RTCP protocol. In each retransmission request, the GOV sequence number and its RTP packet sequence number are defined as the fields of an RTCP packet (a user application RTCP packet). Upon receiving such an RTCP request, the streaming server 11 will try to retrieve the designated GOV from the data link buffer 24 as long as the designated GOV is still there without being overwritten by a new GOV. For an RTCP request for a whole GOV, the streaming server 11 will try to push the designated GOV to the client 12 as soon as possible in multiple RTP packets. For an RTCP request for a single RTP packet, the streaming server 11 takes out the corresponding chunk of data to the client 12 in one RTP packet as soon as possible. In the event that multiple RTP packets of one GOV are lost, the retransmission requests of those RTP packets will be issued one by one. In the case where the designated GOV is no longer existing in the data link buffer 24, that is, in the case where the designated GOV has been overwritten by a new GOV, the streaming server 11 will reply a retransmission-forbidden message to the client 12 through an RTCP packet. Once the client 12 receives such a reply, it marks up the corresponding GOV in the data link buffer 28 with a retransmission-forbidden flag indicating that retransmission is failed and no retransmission request should be re-issued. A retransmission RTP packet is processed as same as a normal RTP packet is. The retransmitted data will be inserted into the corresponding GOV in the data link buffer 28 as long as it is still waiting for the decoder's taking out. The data link buffer 28 checks for any GOVs needing to be retransmitted. The checking process is triggered by the insertion of a new fully recovered GOV. Furthermore, in order not to affect the normal transmission too much, it is preferable to limit the number of retransmission requests for the same GOV data to less than a predetermined number in the case where a retransmission packet is repetitively lost. The predetermined number corresponds to, for example, n times (a predetermined number of times). This limitation is introduced also since retransmission will consume the network bandwidth.

[0128] At the client 12, RTP packets (RTP/UDP packets) that contain GOV data are reassembled into a whole GOV by the help of the reference numbers in each RTP packet, that is, TxGOVSeqNum, Cr, and Tr (see FIG. 8). Because some RTP/UDP packets may be lost or arrive at the destination by out-of-sequence, the RTP/RTCP transport engine client 26 is designed to handle the proceeding problems. There are three types of RG (RTP GOV), that is, new RG, middle RG, and last RG (end RG), where new RG is the first segment of the GOV, last RG (end RG) is the last segment of the GOV, and the rest RGs are middle RGs.

[0129] FIG. 12 shows the structure of a complete GOV with RG (RTP GOV). As shown in FIG. 12, head and end portions of the complete GOV are occupied by a new RG and a last RG (an end RG) respectively. The intermediate portion of the complete GOV is occupied by middle RGs sandwiched between the new RG and the last RG.

[0130] There are three cases for an RTP packet arriving at the client 12, that is, (1) the RTP packet belonging to the current expected GOV, (2) the RTP packet belonging to the GOV that should have been received before the current expected GOV, and (3) the RTP packet belonging to the GOV that should be received after the current expected GOV. Correspondingly, as shown in FIG. 13, there are three definitions, that is, current-in-sequence RG, lagging RG, and leading RG.

[0131] FIG. 14 is an operation flow diagram showing the processing operation of the RTP/RTCP transport engine client 26 which is implemented according to the corresponding segment of the control program for the computer in the client 12. The processing operation of the RTP/RTCP transport engine client 26 in FIG. 14 is executed upon the reception of every RTP packet.

[0132] With reference to FIG. 14, a first step S30 extracts the GOV information from the current RTP packet. A step S31 following the step S30 decides whether or not the current RTP packet is a retransmission packet by referring to the extracted GOV information. When the current RTP packet is a retransmission packet, the step S31 is followed by a step S32. Otherwise, the step S31 is followed by a step S33.

[0133] The step S32 extracts retransmitted data from the current RTP packet, and inserts the retransmitted data into the data link buffer 28. The step S32 is followed by a step S34 for receiving a next RTP packet.

[0134] The step S33 decides whether or not the current RTP packet is a new RG by referring to the GOV information. When the current RTP packet is a new RG, the step S33 is followed by a step S35. Otherwise, the step S33 is followed by a step S36.

[0135] The step S35 decides whether or not the current RTP packet belongs to the current expected GOV, that is, whether or not the current RG is a current-in-sequence RG, by referring to the GOV information. When the current RTP packet belongs to the current expected GOV, that

is, when the current RG is a current-in-sequence RG, the step S35 is followed by a block S37 assigned to a case 1. Otherwise, the step S35 is followed by a step S38.

In paragraphs [0114]-[0115], *Bo* only discloses the normal data transmission between the server RTP/RTCP transport engine 22 and the client RTP/RTCP transport engine 26, as shown in Figure 5. In particular, *Bo* discloses how the transport engine server 22 sets the push timer 33 in response to the RTP packet duration as calculated based on the GOV bitrate and size and the RTP package size. When the push timer 33 triggers, an extended RTP packet (a fragment of GOV) is sent from the RTP/RTCP transport engine server 22 to the client 12.

In paragraph [0116], *Bo* discloses the data retransmission between the two RTP/RTCP transport engines 22 and 26, as shown in Figure 6. These cited paragraphs have **nothing** to do with a clock shift amount as a parameter defined in the client so as to allow the server to adapt the data amount to the reception amount.

Paragraph [0128] describes the different RTP GOVs. Paragraph [0129] describes the structure of a complete GOV with RTP GOV, as shown in Figure 12. Paragraph [0130] describes three different situations in which an RTP packet arrives at the client. Paragraphs [0131]-[135] describe the processing operation of the client RTP/RTCP transport engine as illustrated in Figure 14.

The Examiner fails to point out in which paragraph does *Bo* disclose that the parameter comprises a clock shift amount for preventing playout disruption in the client.

(vi) *Bo* fails to disclose that the adapting comprises an adjustment of a sampling rate

In rejecting claims 8 and 9, the Examiner states that *Bo* discloses that the adapting comprises an adjustment of a sampling rate ([0194], [0195]).

In the following paragraphs, *Bo* discloses:

[0194] FIG. 20 is a time sequence diagram showing the bitrate control message flows. With reference to FIG. 20, the MPEG-4 decoder 13 feeds the data link buffer 28 with a request for reading a GOV. In response to the request, the data link buffer 28 searches for the requested complete GOV. The data link buffer 28 updates the bitrate control information. The data link

buffer 28 sends the bitrate control information to the bitrate adapter module 27 in the client 12. The data link buffer 28 returns the requested complete GOV to the MPEG-4 decoder 13. The bitrate adapter module 27 backs up the last state. The bitrate adapter module 27 makes a decision about bitrate change, and generates a corresponding bitrate change command (bitrate control command). The bitrate adapter module 27 sends the bitrate change command to the bitrate adapter module 23 in the streaming server 11. The bitrate adapter module 23 passes the bitrate change command to the controller in the MPEG-4 encoder 10. Upon the reception of the bitrate change command, the controller in the MPEG-4 encoder 10 returns an acknowledgement to the bitrate adapter module 23 in the streaming server 11. The bitrate adapter module 23 passes the acknowledgement to the bitrate adapter module 27 in the client 12. The bitrate adapter module 27 feeds the data link buffer 28 with a command to change a sliding window. The bitrate adapter module 27 updates the state.

[0195] FIG. 21 is a time sequence diagram showing the retransmission message flows. With reference to FIG. 21, the RTP/RTCP transport engine client 26 inserts a GOV into the data link buffer 28 in the client 12. The data link buffer 28 collects the bitrate control information. The data link buffer 28 searches for a retransmission-permitted GOV (for example, an incomplete GOV or a blank GOV) therein. When the retransmission-permitted GOV is found, the data link buffer 28 sends a corresponding retransmission request to the RTP/RTCP transport engine client 26. The RTP/RTCP transport engine client 26 analyzes the retransmission request and thereby verifies the sequence numbers for both the GOV and the related RTP packet (or packets). The RTP/RTCP transport engine client 26 passes the retransmission request to the RTP/RTCP transport engine server 22. The RTP/RTCP transport engine server 22 passes the retransmission request to the data link buffer 24 in the streaming server 11. The data link buffer 24 verifies the availability of the requested data (the data to be retransmitted). The data link buffer 24 sends either the retransmitted data or a retransmission-forbidden notice to the RTP/RTCP transport engine server 22. The RTP/RTCP transport engine server 22 passes the retransmitted data or the retransmission-forbidden notice to the RTP/RTCP transport engine client 26 by use of an RTP packet. The RTP/RTCP transport engine client 26 extracts the GOV fragment data from the RTP packet, and inserts the GOV fragment data into the data link buffer 28 in the client 12. Alternatively, the RTP/RTCP transport engine client 26 passes the retransmission-forbidden notice to the data link buffer 28. The data link buffer 28 updates the related GOV status.

Paragraph [0194] shows the time sequence for the bitrate control message flows as shown in Figure 20. Paragraph [0195] shows the time sequence in the retransmission message flow. These paragraphs have **nothing** to with adjusting the sampling rate.

For the above reasons, *Bo* does not anticipate claims 2-9. For the same reasons, *Bo* fails to anticipate the dependent claims 10, 12-20, 22-25, 27-31 and 33-36.

In sum, *Bo* fails to disclose that 1) the client sends to the server information indicative of the parameter defined by the client for determining a rate adaptation range between the server and client and 2) the client adjusts packet transfer delay variation. Thus, *Bo* fails to anticipate independent claims 1, 11, 21, 26 and 32.

As for dependent claims 2-10, 12-20, 22-25, 27-31 and 33-36, they are dependent from claims 1, 11, 21, 26 and 32, and recite features not recited in claims 1, 11, 21, 26 and 32. For reasons regarding claims 1, 11, 21, 26 and 32 above, claims 2-10, 12-20, 22-25, 27-31 and 33-36 also distinguishable over the cited *Bo* references.

Furthermore, regarding claims 2, 12, 22 and 29, *Bo* fails to disclose that the parameter comprises a minimum shift amount indicative of a difference between a sampling time and a transmission time of a packet at the server.

Regarding claims 3, 13, 23 and 30, Bo fails to disclose that the parameter comprises a target shift amount indicative of a shift amount greater than a difference between a sampling time and a transmission time of a packet at the server.

Regarding claims 4, 14, 24 and 31, Bo fails to disclose that the parameter comprises a number specifying a maximum difference between the number of bytes than has been sent and the number of bytes that have been sampled.

Regarding claims 5 and 15, Bo fails to disclose adapting a sampling rate to the transmission rate in the server based on the parameter.

Regarding claims 6, 16 and 25, *Bo* fails to disclose that the parameter comprises a clock shift amount for preventing playout disruption in the client.

Regarding claims 8, 9, 18, 19, 35 and 36, *Bo* fails to disclose that the adapting comprises an adjustment of a sampling rate.

CONCLUSION

Claims 1-36 are allowable. Early allowance of all pending claims is earnestly solicited.

Respectfully submitted,

Ken Lao

Kenneth Q. Lao
Registration No. 40,061

Date: February 15, 2007

WARE, FRESSOLA, VAN DER SLUYS
& ADOLPHSON LLP
Bradford Green, Building 5
755 Main Street, PO Box 224
Monroe, CT 06468
(203) 261-1234